

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 1 065 470 A1

(12)

## EUROPEAN PATENT APPLICATION

(43) Date of publication:  
03.01.2001 Bulletin 2001/01

(51) Int Cl.7: G01C 21/00, G08G 5/04

(21) Application number: 99112442.1

(22) Date of filing: 30.06.1999

(84) Designated Contracting States:  
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE  
Designated Extension States:  
AL LT LV MK RO SI

(72) Inventor: Min-Chung, Gia  
Nei-hu, Taipei (TW)

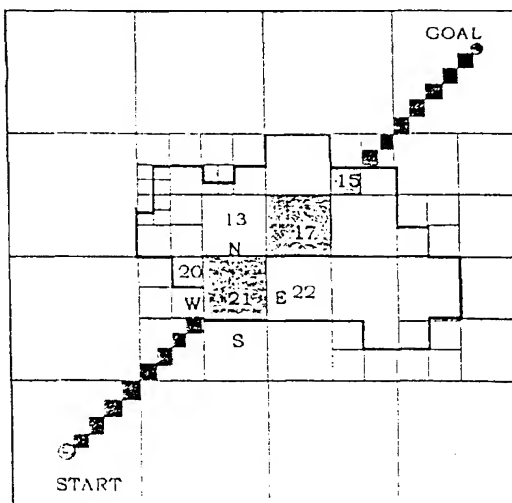
(74) Representative:  
Patentanwälte Rüger, Barthelt & Abel  
Postfach 10 04 61  
73704 Esslingen a. N. (DE)

(71) Applicant: Min-Chung, Gia  
Nei-hu, Taipei (TW)

(54) A path planning, terrain avoidance and situation awareness system for general aviation

(57) A number of navigation functions are performed on terrain navigation space. A preferred embodiment of dynamic dangerous zone defined by flight altitude is demonstrated. In a preferred embodiment, a set of nodes of terrain height over a minimum flight altitude are located and aggregated. Algorithms such as collision check, mountainous area boundary and region growing technique are developed as basic operations for this terrain model. Yet another preferred embodiment with visibility graph approach for dynamic route selec-

tion has been adapted to reduce the real-time computational requirements. This approach reduces the size of the search space by establishing a partial visibility graph of terrain and avoids details of the terrain, which do not influence the choice of flight path, independent of the size of the navigation space. By exploiting the multiple and variable resolution properties of Oct-tree terrain models, a series of CFIT warning functions using terrain data as reference are implemented efficiently with existence terrain data resource on board.



(b) --- LINE ELEMENTS OF DIRECT PATH.  
--- COLLISION NODES.

Figure 3. An example of navigation space with danger nodes.

## Description

### BACKGROUND OF THE INVENTION

#### FIELD OF THE INVENTION

[0001] This invention generally relates to a system for reducing controlled flight into terrain accidents and in particular it relates to terrain data processing and algorithms in terrain awareness and warning systems of controlled flight into terrain accidents.

#### BRIEF DESCRIPTION OF THE PRIOR ART

[0002] Controlled Flight into Terrain (CFIT) warning uses the data provided by flight management computer, the Global Positioning System (GPS) receiver and other aircraft substances. The prior art of CFIT warning predicts a three-dimensional flight path based upon a curve-fit extrapolation of the most recent position and velocity data received. This predicted flight path is then compared with the internal terrain map of the immediate area and an assessment of the potential threat of aircraft collision with terrain is computed. The type of systems described above could look up to 120 seconds in advance. This is so called "time to clear" in contrast to "time before impact" warning, and also "terrain avoidance" in contrast to terrain "awareness".

[0003] The effort to reduce CFIT accidents by U.S. Airlines can be traced back to 1974. The prior art of standard Ground Proximity Warning System (GPWS) use radio altimeter data to provide an audible warning if an aircraft has insufficient terrain clearance. Flying into precipitous terrain can result in late warnings since the standard GPWS depends upon downward looking radio altimeter to detect rising terrain.

[0004] "Enhanced" version of GPWS - EGPWS, which is available from AlliedSignal, and Ground Collision Avoidance System (GCAS), which is available from Sextant Avionique, graphically depict terrain surrounding the aircraft's flight path on a cockpit display and provide earlier warning. Both systems are built around a three-dimensional terrain database and implement a true predictive look-ahead capability that is based on aircraft climb performance. The technology has three parts: the advent of GPS and other highly accurate navigation system to provide precise positioning with updates in seconds; computer technology with greater speed and memory; and accurate, sophisticated worldwide terrain databases including a Digital Terrain Elevation Database (DTED) of the world. Figure 1 shows schematic system diagram of GPWS, which is from DASSAULT ELECTRONIQUE GCAS.

[0005] In EGPWS and GCAS, the terrain database is the system's core. Looking-ahead algorithms compare the projected future position of the aircraft with this database and warnings are issued on this basis. The using of DTED allows the systems to display terrain in prox-

imity to the aircraft. During the flight, the peaks and terrain below the aircraft in DTED can also be displayed for situation awareness to the pilot. Many of looking-ahead warning such as flight profile or flight path to be in conflict with the terrain can be given in advanced alert. During emergency descents in mountainous area and during en-route avoidance of inclement weather, looking-ahead warning further helps to prevent any situation that could lead to a CFIT.

[0006] However, terrain database occupies large amounts of memory. If the terrain is mapped at 100-meter intervals over a region 10000Km by 10000Km, there are  $10^{10}$  grid points. Clearly, the computations inherent in continuously accessing  $10^{10}$  grid points are formidable and it is necessary to reduce, or compress this information. Besides, many of navigation functions can be done with knowledge of terrain, such as allowing for optimum flight plan or emergency change of route in real time situation, which could be restrained by the number of data retrieving and the cost of computation when complexities of algorithms increased. Another major drawback of DTED file is that it only provides elevation data. Without further processing or aids from other formats of terrain data, such as features or vector representations, DTED gives no geometrical relationship among data elements.

#### SUMMARY OF THE INVENTION

[0007] There are many applications, particularly in flight management, where the complete DTED database must be accessed to determine or revise the flight waypoints. The terrain awareness warning and navigation safety related issues described here are performed on an encoded terrain navigation space. The encoding of grid file of Digital Terrain Elevation Data (DTED) is based on a variant of quad-tree representation of spatial data structure. Each element of a DTED file are encoded in a Morton numbering sequence with respect to its location in the grid file, a scaled elevation data and coverage of homogeneous (equal elevation) area as features. Under this data structure, a DTED is organized as a set of integer numbers with ascending sequence. Each integer represents a node in which planar location, scaled elevation and coverage are interlaced to a set of bits position to form an integer. The encoded list is defined as terrain Oct-tree model. Navigation functions not only make reference on terrain Oct-tree for elevation data, but also perform processing and operation on it. The navigation space is transferred from DTED array to its encoded integer list.

[0008] A number of navigation functions are performed on terrain Oct-tree based navigation space. A preferred embodiment of dynamic dangerous zone defined by flight altitude is demonstrated. In a preferred embodiment, a set of nodes of terrain height over a minimum flight altitude are located and aggregated. Algorithms such as collision check, mountainous area

boundary and region growing technique are developed as basic operations for this terrain model. Yet another preferred embodiment with visibility graph approach for dynamic route selection has been adopted to reduce the real-time computational requirements. This approach reduces the size of the search space by establishing a partial visibility graph of terrain and avoids details of the terrain, which do not influence the choice of flight path, independent of the size of the navigation space.

**[0009]** Several forms of CFIT warning functions of aircraft navigation which become feasible once an aircraft flight path and the topology of a region of terrain can be readily determined. Furthermore, it implies access to a database holding the terrain data in order to initiate the geometric computations. By exploiting the multiple and variable resolution properties of Oct-tree terrain models, a series of CFIT warning functions using terrain data as reference are easily implemented. The functions including Ground Proximity Warning, Obstacle Cueing, Terrain Masking, Perspective Images of Terrain, Passive Ranging, Real-time Route Selection and Route Planning, Weather Display Overlaying, and Waypoint Overlaying.

**[0010]** Prior art planning approaches use pre-defined obstacle models. Moreover, the number of data retrieving DTED and the cost of computation are in the negative. In present invention, the dangerous zones dynamically vary during the execution of a flight plan. Besides, a layer of Oct-tree terrain is used to facilitate the on-line operation capability.

**[0011]** Other objects and advantages of the present invention will become apparent from the detailed description to follow taken in conjunction with the appended drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0012]** In the drawings:

**[0013]** Figure 1 shows a schematic system diagram of a prior art of GPWS.

**[0014]** Figure 2 shows bit interleaving of locational code and 2-D, 3-D projection.

**[0015]** Figures 3(a)-3(b) show an example of navigation space with danger nodes.

**[0016]** Figure 4 shows possible dangerous zones with respect to heading.

**[0017]** Figures 5(a)-5(b) show an example of using binary search of danger node list to determine the boundary nodes during dangerous zone expansion (dangerous zone refer to Figure 3).

**[0018]** Figure 6 shows a combination of boundary type and waypoint position.

**[0019]** Figure 7 shows dangerous zone expansion and waypoints derivation.

**[0020]** Figures 8(a) - 8(d) show dangerous zones with respect to different start and goal point.

**[0021]** Figure 9 shows a visibility graph and its tree structure.

**[0022]** Figure 10 shows an example of embodiment of real time flight path planning.

**[0023]** Figure 11 shows an example of embodiment of obstacles avoidance.

**[0024]** Figure 12 shows an example of embodiment of perspective image generating.

## DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

### 1. Terrain model

**[0025]** In the image processing field, a quad-tree can represent a two dimensional region (in a  $2n \times 2n$  binary array format) by recursively subdividing the array into quadrants. If the quadrant consists of a mixture of 1s or 0s, it is further sub-divided into quadrants and this process is repeated until a quadrant consists only of 1s or 0s (termed leaf node). In practice large region of 1s (or 0s) are represented by a single quadrant or node of the tree. Similarly, three-dimensional objects can be represented by Oct-trees where a  $2^n \times 2^n \times 2^n$  array is sub-divided into octants. If the elements of an octant are common, the Oct-tree terminates; otherwise eight further sub-octants are generated to represent the octant in more detail.

**[0026]** In the present preferred embodiment, a variant of Oct-trees structure is adopted. A fourth parameter S is added to represent the size of a leaf node. A single 3-D locational code can represent an area of terrain surface by its three-dimensional co-ordinates and size information. The method of mapping a peak point (I, J, K) of terrain surface to the locational code of a node is formed by bit interleaving the triplet (I, J, K) and S as a hexadecimal value. Such that the S bit precedes the K bit, the K bit precedes the J bit, and the J bit precedes the I bit at each position. After interleaving (I, J, K) and S, the locational code is one of the set of digits {0,1,2,3,4,5,6,7,8,C}. The times of digit '8' or 'C' appears in the location code gives the size information of a merged node.

**[0027]** In the example of figure 2, the combination of the  $4^{n-1}$ th bits (i.e., the combination of the first bit of each digit) of a node is equal to the binary representation of value I. This is repeated for the  $(4^{n-1}+1)$ th bits to J, the  $(4^{n-1}+2)$ th bits to K and the  $(4^{n-1}+3)$ th bits to form the size value S respectively, where n ( $n=1,2,\dots$ ) is the resolution parameter. Although the four components of a node (i.e., I, J, K, S) are totally independent, they are represented by a single value. Both the encoding and decoding process can be performed by the use of bit-wise operations under this bit interleaving scheme.

**[0028]** The addressing schemes used in terrain Oct-trees are Morton numbering sequence, which are formed by bit interleaving as previously described. The 2-D locational code can be obtained by the use of modulo-4 operations to the I, J bits of the 3-D locational code, where the K bits are assigned to 0, and the S bits are

left unchanged. In Figure 2, the projection of a node <00001473><sub>16</sub> on the IJ plane is <00001033><sub>16</sub>; for the merged node <0000148c><sub>16</sub> the projection code is <00001088><sub>16</sub>.

## 2. Modeling of the Navigation Space

**[0029]** In general, existing methods of path planning consist of two phases. In the first phase, a search space is generated consisting of all the possible paths that will avoid the obstacles between a start point and a goal point. Once the search space is created, the second phase is to search for a path in which satisfies specific constraints. These approaches are typically based on the assumption that the navigation space (obstacles, terrain, and threat) is static and is fully known a priori.

**[0030]** Many prior art flight path planning algorithms exploit knowledge of the 'cost' of paths in the navigation space to extract an optimal path by minimizing some objective function which defines the 'cost' of a path. These algorithms are variants of the 'shortest path problem'. A common strategy is the pre-processing of the search space to reduce the real-time computation cost. The pre-processing approaches can be classified in two categories, one which organizes the graph of the search space according to the terrain altitude and the other which embeds pre-computed cost values in a graph of the search space. The flight path planning algorithm described in this invention is based on constraints of flight altitude, flight distance and time to goal. Although the aircraft knows the navigation space, the obstacles vary with operational constraints and are therefore explored in real-time, during the mission.

**[0031]** The flight path planning algorithm described in the present invention is performed on a terrain Oct-tree representation of the navigation space and without the use of any other data formats to represent terrain features (linear or polygonal features). Since the set of obstacles dynamically change with flight altitude, a static polygonal representation of obstacles is not suitable for real-time dynamic flight path planning. The terrain Oct-tree is organized as a linear list in which each element represents a leaf node of the tree. Each node is represented as a single integer and represents a homogeneous set of elements of the grid file. The use of an Oct-tree simplifies the extraction of obstacles from a terrain database.

**[0032]** An Oct-tree terrain model contains both two-dimensional (2-D) and three-dimensional (3-D) information. However, the method of tree encoding allows straightforward one-to-one projections between 2-D quad-tree space and the corresponding node in 3-D Oct-tree space. This allows a 3-D flight path to be computed in quad-tree space, considerably simplifying the computation task. The use of 2-D representations also simplifies such tasks as the location of neighboring points, determination of a path segment between two points and transformations between geographic co-or-

dinate and node addresses.

## 3. Core Path Planning Algorithm

### 3.1 The Extraction of Obstacles

#### 3.1.1 Exploring the Obstacles

**[0033]** The nodes in a terrain Oct-tree with scaled elevation values above a flight altitude are known as "danger nodes", in the sense that an aircraft cannot safely enter a region of the terrain occupied by these nodes. The danger node list is a sub-set of the quad-tree representing the nodes of the navigation space. However, the list of danger nodes does not give any explicit topographic information (for example, connectivity or boundary conditions) in the navigation space. In Figure 3, a danger area containing 26 nodes is shown, organized as an ordered list where it is not clear if node 1 or node 26 belong to the same connected region. Moreover, although the danger nodes are 'scattered' throughout the navigation space, only a few of the danger nodes will jeopardize a specific flight path. For instance, in Figure 4, there are five connected danger regions but, for a path to goal 1, the possible obstacles are limited to regions A and C.

**[0034]** The danger nodes related to the current direction are termed obstacle nodes and are organized as a set of locational codes, which represent the coverage of dangerous zones in the aircraft navigation space. The organizing of obstacles involves the following steps:

(1) A list of danger nodes is extracted from the terrain Oct-tree according to an elevation threshold value. The terrain is organized as bands or layers with a suitable vertical scale factor K, for example, 20m bands. Each node of the Oct-tree is examined to see if its K value (embedded in the locational code) exceeds the threshold. Nodes with K value less than the threshold can be ignored. The resultant list contains all the potential danger nodes in the navigation space.

(2) Using a 'generate and test' paradigm, the obstacle nodes list is obtained by collision check (or intersection detection) of the direct path which lies between a given start point (S) and a goal point (G). The direct path is a line between points S and G to determine the Oct-tree nodes on a straight line between the points S and G. Each point along the line, beginning at point S, is checked against the list of danger leaf nodes. If a point on the line matches any danger node, then the danger leaf node is stored in a list for subsequent expansion of the dangerous zone. Sometimes, more than one component of a point along a direct path will intersect with the same obstacle node.

(3) A connected dangerous zone is grown to locate the vertices of the region as waypoints. A danger-

ous zone expansion process involves finding the obstacle leaf nodes adjacent to the leaf node being expanded. The major purpose of this expansion process is to obtain a set of waypoints corresponding to the dangerous zone in navigation space; these waypoints are subsequently used as possible flight path deviations to avoid a collision with the terrain.

### 3.1.2 Dangerous zone Expansion

**[0035]** Computing the locational codes of its neighbors in four main directions expands the obstacle leaf node. This operation is performed recursively until a boundary node is reached which does not have any neighboring nodes in the list of danger leaf nodes. The size of a neighboring node may be different from the size of a current 'expanding' node, where the size of a  $2^d$  by  $2^d$  node is  $2^d$  and the level of the node is defined as  $d$ . The danger node list is searched for a neighbor of equal size. If an equal size neighbor is not found in the list, the search is repeated for a neighbor with a larger size and this process repeats until a neighbor is found or the tree level next to the root is reached.

**[0036]** If a neighbor node is still not found after this process, it indicates that either some smaller size neighboring nodes exist or the node is a boundary node. If the node is not present in the node list, it may be contained in a merged node in an upper level. The node with the projection code which is used in the last comparison loop of a binary search procedure actually covers or is covered by the query node. This feature can be exploited to determine if any quadrants of the same size neighboring node exist. If the neighboring node of the node under test covers the locational code, which is used in the last comparison loop, a further search is necessary, otherwise the current expanded node is a boundary. For example, the expansion process of node 21 in Figure 3 encounters a boundary in the 'south' direction, connects obstacle nodes in the north and east directions, and needs further processing to the next level in the 'west' direction as shown in Figure 5.

**[0037]** The boundary type of an obstacle is encoded using the additive codes 1, 2, 4 and 8 correspond to the sides on the northern, eastern, southern and western boundaries, respectively. Zero indicates that a node that does not have any side that is on the boundary. Figure 6 shows a dangerous zone and the potential positions of waypoints. For example, a node with a boundary code of 13 is a vertex node. It has two waypoints in the NW and SW diagonal direction. A waypoint is represented by the locational code of the northwest corner of its neighbors in the diagonal direction. It is also necessary to check that a waypoint is not a member of the list of danger nodes and does not already exist in the waypoints list, before it can be appended to the list of waypoints.

**[0038]** Two further approaches can be applied to re-

duce the size of expansion node or to use a coarser level node approximation. The first approach is implemented when more detailed information is required. It is achieved by subdividing the currently expanded node and expanding each sub-quadrant individually. In this case, the currently expanded node is treated as a boundary node and its boundary type is updated. This process is performed recursively until either a neighboring node is found or the process reaches a boundary node. In the second method, the hierarchical features of a terrain Oct-tree are exploited. In order to reduce the number of vertices (and hence reduce the size of the visibility graph), rather than the size of the currently expanded node, the locational code of an equal size neighboring node is assigned as a boundary node and the expansion process is terminated. This is an approximation process that 'truncates' nodes below the current expanded resolution level. The boundary code of a currently expanded node is updated as described above.

### 3.1.3 Waypoint Location

**[0039]** Whenever a neighbor is located, it indicates that further expansion is needed in that current direction, otherwise the node has encountered a boundary node. After all the four main directions are examined, the boundary type of an obstacle node is obtained. From the boundary type, it is straightforward to determine if the node is an obstacle node. The expansion process proceeds recursively from the neighboring nodes and an obstacle nodes list is used to refer to nodes which have been expanded, in order to avoid examining the same obstacle nodes repeatedly. Figure 7a shows the expansion of a dangerous zone. During this expansion process, the boundary type of a boundary node is obtained in order to determine if it is a vertex of a dangerous zone. If it is a vertex node, a waypoint is then created in the 'diagonal direction' adjacent to the dangerous zone, as shown in Figure 7b.

**[0040]** After the expansion process has been applied to all the members of the SEED list, the dangerous zones are obtained along a direct flight path between the start and goal point, which is dependent on the overall direction of the flight path. Figure 8a-d shows a gaming area with danger nodes and corresponding regions of obstacles along the different direct paths between the start and goals nodes. The actual topography information of the dangerous zones is extracted from the specific terrain Oct-tree and is transformed into a set of waypoints. These waypoints are then used to construct a visibility graph in the navigation space, to determine the optimal path.

### 3.2 Transformation of Navigation Space

**[0041]** The construction of a visibility graph is based on a set of waypoints obtained during the acquisition stage. This set of waypoints provides the implicit geo-

graphic information of the dangerous zones in navigation space. The algorithm consists of considering all pairs of points ( $W_{\text{from}}$ ,  $W_{\text{to}}$ ), where  $W_{\text{from}}$  and  $W_{\text{to}}$  are start, goal, or intermediate waypoints of dangerous zones. To determine whether  $W_{\text{from}}$  and  $W_{\text{to}}$  are the endpoints of a valid flight path segment, 'collisions' are checked against dangerous zones of a straight line  $W_{\text{from}}$  and  $W_{\text{to}}$ . The nodes between  $W_{\text{from}}$  and  $W_{\text{to}}$  are connected by a link in the visibility graph if, and only if, no intersection occurs in the segment joining the two points.

**[0042]** The criteria for collision checking is similar to the one used to check for collisions of direct paths between start and goal points. The test of a pair of waypoints is terminated as soon as a collision is detected, otherwise the test proceeds until  $W_{\text{to}}$  is reached. After all the possible combination of waypoint pairs are tested, the result is a visibility graph where the waypoints are the nodes of a graph and the path segments formed by the waypoints are the arcs of a graph. In Figure 9, several paths exist which are composed of straight lines joining the start point to the goal point via a sequence of waypoints.

**[0043]** The visibility graph approach used is similar to the methods used in most paths planning problems where a visibility graph of the obstacle space is constructed from a list of polygonal obstacles. However, the following considerations are specific to terrain Oct-tree based flight path planning algorithm:

**[0044]** The dangerous zones are obtained by gathering the obstacle nodes along an ideal direct path between the start and goal points and the possible obstacles are limited to those nearby the direct path (or current direction). This approach implies that the irrelevant danger nodes in the navigation space can be ignored and the number of waypoints is accordingly minimized. Because the waypoints only represent the subset of obstacles of the danger areas, this process provides a partial visibility graph of the total navigation space. Figure 8a-d shows an example of partial configuration and its visibility graph.

**[0045]** During the collision check,  $W*(W-1)/2$  waypoint pairs where  $W$  is the number of waypoints, including the start point  $S$  and goal point  $G$ . The time complexity for construction of a visibility graph is proportional to  $W*(W-1)/2$ . For any navigation space, it can be readily observed that the number of waypoints  $W$  is less than the number of vertices  $n$ . Moreover, the size of a visibility graph containing  $W$  waypoints is smaller than the size of a visibility graph of  $n$  vertices. Clearly, significant improvements in the speed of searching can be obtained possible by reducing the size of a visibility graph.

### 3.3 Flight Path Searching

**[0046]** The visibility graph is represented in a form of a list of flight path segments as described in the previous section. The path planning problem has been trans-

formed into the discrete problem of searching a visibility graph between a start node and a goal node. For example, Figure 9 shows a visibility graph representing path segments of a navigation space. There are seven nodes in the graph including the start and goal points. It is reconstructed as a tree structure where the goal node  $G$  and other nodes appear more than once in the tree to illustrate its construction.

**[0047]** Several techniques can be used for searching for path in a graph, including such as depth-first, breadth-first and heuristic search. A heuristic search method, constrained to minimize the number of connections has been incorporated in the flight path planning algorithm described in preferred embodiment. During the search, a path segment is chosen that is as distant as possible from the current position, in the expectation that it will be closer to the destination. On the other hand, the method of minimizing the length of the flight path entails an opposite approach to the search to minimize the connections. Cost functions are defined to evaluate the effect of adding waypoints to determine which set of waypoints should be expanded in the path planning routine.

**[0048]** Finding a truly optimal path would require an exhaustive search. However, an optimal path may not be essential to real-time airborne requirements because of the possible degradation in performance to extract the path. For a large number of nodes in a visibility graph, the computation load of an exhaustive approach is prohibitive, and a more efficient heuristic search methods was adopted, based on a variant of Dijkstra's algorithm, which is also known as the  $A^*$  method.

### 4. Real Time Dynamic Environment Application

**[0049]** As it is anticipated that an aircraft flight path needs to be modified in real-time to match flight conditions and changing obstacles in the environment, the terrain database is accessed for every flight plan change in order to re-form the visibility graph, based on the current obstacle space. This re-planning of a new flight path must be completed within a few seconds for real-time navigation, where this time interval includes the flight path planning computation and also the construction of the path searching space.

**[0050]** Typically, in a real-time navigation environment, a new flight path is required within few seconds of a request. In practice, extracting waypoints and searching the visibility graph take a relatively small amount of time in comparison with the time taken for the whole path planning process. The strategy of applying the proposed path planning algorithm to the real time dynamic environment is based on observations from applying the algorithm to a specific terrain with randomly generated start and goal points and measuring the time spent on each stage. These measurements include the time to generate the waypoints, the time to construct a visibility graph and the time to find a path at a different

resolution level of the terrain Oct-tree. These off-line results of the computation times are subsequently used as references for 'tuning' the real time dynamic flight path planning algorithm for a given terrain.

**[0051]** To vary the resolution of the navigation space, a pyramid of quad-trees is used to represent the navigation space and danger nodes. A layer  $k$  of a pyramid is obtained by applying a maximum value function on a  $2 \times 2$  window at layer  $k+1$ . However, at upper layers of a pyramid, the path may be obscured as a result of 'rounding' the tree nodes and consequently, no path may be found at that layer. In real time applications, it is preferable to avoid the bottleneck cause by establishing a large visibility graph at a detailed resolution layer. On the other hand, it is also important to avoid a coarser resolution layer, which effectively hides the valid path. The determination of the appropriate processing level can be accomplished by first obtaining the waypoints at a pre-defined level, estimating the size of visibility graph and then determining if it is necessary to switch to another layer for path planning.

**[0052]** For example, Figure 10 demonstrates a real time simulation by continuously changing the goal point during a flight mission, in which the initial start point is (255,101) and the goal point is (96,251). Airspeed of 400 m/sec is predefined and the flight altitude and operation layer is set interactively by the user. During the flight, a new goal point (241,242) is given for new flight path planning; the algorithm predicts the new start point as (204,148) according to a 5-second constraint. The path is found in the same layer and flight altitude as shown in Figure 10. Different baselines, scaling factors, time constraints and flight altitude are also applied.

**[0053]** The DTED source file is 1:50000 Scale Height Data termed Digital Terrain Model Data (DTM) provided by the Center for Space and Remote Sensing Research, National Central University, Taiwan ROC. The DTM file consists of height values at each intersection of a 40 meter horizontal grid, with values mathematically interpolated from the contours on the SPOT Landmass maps containing 256K height values for a 20.48 km square DTM 'tile'. The source file can be as large as  $2^{16} \times 2^{16}$  grid points, but has been restricted to  $2^9 \times 2^9$  grid points to simplify the encoding process.

**[0054]** As described above, the path planning algorithm can be performed at a coarser layer of a pyramid to meet real time constraints. However, the terrain elevation data is continuous varying and the terrain Oct-tree representation is terrain dependent, thus the location and connectivity of the obstacles with respect to a given flight altitude are unpredictable. Generally, the actual operation layer of the path planning process is determined by the minimum flight altitude, which in turn determines the number of waypoints. A low flight altitude will generate a large number of danger nodes and obstacle nodes, thus a coarser operational layer may be adopted to keep the number of waypoints below a pre-defined margin for a specific operational environment.

As the real time application depends on the performance of the computing system and the aircraft speed, the real time constraint, the allowable times for the planning and the predefined layer of process are changeable.

## 5. Terrain Avoidance and Situation Awareness

### 5.1 Dynamic Obstacles and Weather Condition Avoidance

**[0055]** The present invention not only provides warnings in accordance with surrounding terrain, but also gives the information of terrain alone flight path ahead, which is obtained from path planning function. Once a new route has been on-line programmed, the warning function of GPWS and GCAS can be easily coupled. Other types of hazards coverage area such as obstacles, peaks, weather conditions (thunderstorms, windshear or any meteorological situations that appear in airspace) can also be represented as a set of locational codes overlaying terrain map and performed in the same way as described above. Figure 11 shows an example of embodiment.

### 5.2 Terrain Map and Terrain Masking for Terrain Awareness

**[0056]** The generating of terrain map and computing of line-of-sight terrain masking of the dangerous zone, peaks, and obstacles with respect to a selected flight altitude and heading use the same locational code representations. An accessing algorithm is used to retrieve parameters of nodes in terrain model. The I, J, K parameters give the planar position and elevation of a node, different color codes from a mapping table are assigned to each elevation band; S representing the coverage area of said node to form said masking area. The subset of nodes representing dangerous zone, peaks, and obstacles is obtained from collision checking and region finding with homogeneous nodes.

### 5.3 Perspective Images of Terrain Model

**[0057]** Method of generating perspective images of terrain model, including the retrieving and accessing data from terrain model for generating images is provided. An accessing algorithm is used to retrieve parameters of nodes in terrain model. The I, J, K parameters give the planar position and elevation of a node, different color codes from a mapping table are assigned to each elevation band, S representing coverage area of a node to form said perspective images of terrain. The terrain model is used directly as input data without accessing the original DTED file. Figure 12 shows an example of embodiment of perspective images.

## 6. Conclusions

**[0058]** Prior art planning approaches use pre-defined obstacle models. However, the dangerous zone obstacle area varies during the execution of a flight plan whenever the flight altitude is changed. Besides, in prior art DTED approach, the number of data retrieving and the cost of computation are in the negative. Optimum flight plan or emergency change of route in real time situation could be restrained by the number of data retrieving and the cost of computation when complexities of algorithms increased in standard DTED based system.

**[0059]** In the present invention, algorithms for CFIT warning based on Oct-tree terrain are presented. It is straightforward to represent terrain at various levels of resolution. An Oct-tree structure has the ability to modeling the terrain with sufficient accuracy at a coarser level by truncating or approximating unnecessary data allowing terrain reference tasks to be performed efficiently. In the present preferred embodiment, by using the features of Oct-tree terrain model, the terrain awareness and warning system related functions are implemented more efficiency than prior art DTED based method. Moreover, comparing with prior art of pre-flight planning system, the preferred embodiment shows a real-time flight path planning capability.

**[0060]** For a long-range global path planning in airborne environment, either in pre-planing or in real-time situation, Oct-tree hierarchical decomposition structure avoids excessive detail of terrain in path planning phase. By exploiting the hierarchical nature of Oct-trees terrain model, the path planning algorithm can operate at any layer of the Oct-tree terrain. Moreover, the path planning approach reduces the size of the search space by establishing a partial visibility graph of the navigation space which do not influence the choice of path and avoids details of the terrain. The Oct-tree terrain and visibility graph approach described in present invention has been adapted to meet the real-time computational requirements.

**[0061]** For CFIT warning related functions and obstacles avoidance, the same scheme as flight path planing can be used. By replacing the DTED with Oct-tree terrain or adding a layer of Oct-tree terrain, the comparing of predicted flight path with terrain database for detecting possible collision and giving warning or for obtaining a new route for avoidance could be easily implemented. Furthermore, the terrain awareness and display can be performed in more efficient way.

**[0062]** Although exemplary embodiments of present invention are described above, this description does not limit the scope of the invention, which can be practiced in a variety of embodiments. For example, the real-time flight path planing algorithm can be embedded in Flight Management Computer (FMC) for flight control. On the other hand, the flight path and terrain profile obtained from real-time flight path planning together with encoded terrain database give the aircraft a forward looking

capability particularly in general aviation when flight in mountainous area and urban area with buildings and man made obstacles. Accordingly, this capability can be easily extend to Controlled Flight Into Terrain Warning, Ground Proximity Warning, and Obstacles Avoidance. Moreover, terrain awareness and display could be one of the features coupled with CDTI.

**[0063]** In the point view of terrain database, with limited cost, the terrain model can be stand alone or parallel with the prior art of DTED database, even to replace the existing DTED type database. In the field of military, mission planning, terrain avoidance, terrain following, terrain matching, radar and threat line-of-sight masking, intervisibility shading are possible implementations. For non-aviation usage such as seafloor representation, sonar coverage, the estimation of GSM base station coverage can also be implemented.

**[0064]** While the present invention has been shown and described with reference to a preferred embodiment thereof, and in terms of the illustrative drawings, it should not be considered limited thereby. One skilled in the art could conceive of various possible modifications, omissions, and alterations and the content of any particular embodiment, without departing from the scope of the present invent.

## Claims

1. A method of performing real-time flight path selection and path planing for general aviation, comprising the following steps:
  - using Digitized Terrain Elevation Data (DTED) to generate a terrain model;
  - using this terrain model to provide a navigation space;
  - accessing and retrieving terrain model to generate a terrain map;
  - giving start and goal points on navigation space to determine a ground track of direct flight path on terrain map;
  - identifying dangerous zone based on ground track and flight altitude;
  - using said dangerous zone to allocate a set of way-points for avoidance;
  - constructing a visibility graph of navigation space, in which is a set of collision free path segments ;
  - linking start point to goal point by flight path searching algorithm; and
  - obtaining the terrain profile of flight path from terrain model.
2. The method in claim 1, wherein said terrain model is an Oct-tree terrain.
3. The method in claim 1, wherein said terrain model



is a variance of quad-tree and Oct-tree structures.

4. The method in claim 1, wherein said terrain model contains a set of nodes, each represented by an integer.

5. The method in claim 4, wherein

each of said nodes contains four parameters I, J, K, S ;  
the four parameters I, J, K, S are obtained by mapping each elevation peak point I, J, K of a DTED file into a 3-D locational code of the corresponding Oct-tree;  
parameters I, J define the 2-D coordinates of planar location, with elevation K, the parameters I, J, K define the 3-D coordinates of a space location;  
scaling factor of the elevation is introduced which divides the terrain elevation into bands; parameter K represents scaled elevation;  
the nodes of said Oct-tree with equal scaled value of elevation occurs in the same quadrant of the corresponding quad-tree are merged;  
scaling factor can also be non-linear, or begin with a baseline in stead of the mean sea level;  
parameter S is the coverage area of a node, and is added to represent the size of a node;  
said four parameters I, J, K, S are interleaving in its bit positions to form an integer representation of node; and  
each terrain elevation data has its unique correspondent node.

6. The method in claim 1, wherein

said navigation space is defined as a region for allocating possible flight path;  
said navigation space combines Oct-trees with quad-trees to provide both 3-D and 2-D operations on terrain elevation data; and  
the 2-D location code can be obtained by removing the K bits from 3-D locational code.

7. The method in claim 5, wherein said terrain map is obtained by the following steps:

using an accessing algorithm to access said terrain model;  
accessing a node means accessing an coverage area of said node;  
using a retrieving algorithm to retrieve parameters of a node of said terrain model, where the I, J, K parameters give the planar position and elevation of a node, different color codes from a mapping table are assigned to each elevation band, S representing coverage area of a node to form said terrain map.

8. The method in claim 1, wherein

the ground track is a straight line segment, formed as a list of nodes;  
the dangerous zone is a list of nodes obtained from collision check in which elevation of the nodes conflict with flight altitude;  
each collision free path segment is a straight line segment, formed as a list of nodes;  
the way-points are represented in 2-D locational codes; and  
the visibility graph is constructed by performing collision check on each segment between pairs of way-points to determine a set of collision free path segments.

9. The method in claim 1, wherein  
due to the geometric region features of said dangerous zone in navigation space, path searching algorithms based on visibility graph are used.

10. A method of performing real-time dynamic collision check comprising the following steps:

using Digitized Terrain Elevation Data (DTED) to generate a terrain model;  
using this terrain model to provide a navigation space;  
accessing and retrieving terrain model to generate a terrain map;  
identifying a list of nodes of on navigation space based on flight altitude;  
giving a ground track of flight path segment which is a list of nodes on terrain map; and  
determining the flight path segment in conflict with dangerous zone by searching each nodes along the path segment against the list of nodes of dangerous zone.

11. The method in claim 10 wherein the collision check is performed by checking common nodes appearing in both node lists.

12. A method of performing real-time dynamic weather condition avoidance comprising:

using Digitized Terrain Elevation Data (DTED) to generate a terrain model;  
using this terrain model to provide a navigation space;  
accessing and retrieving terrain model to generate a terrain map;  
giving start and goal points on navigation space to determine a ground track of direct flight path on terrain map;  
identifying dangerous zone based on ground track and flight altitude;  
giving a weather condition coverage area which

is represented by a list of nodes;  
 adding the list of nodes of the weather coverage  
 area to the list of nodes of the dangerous zone;  
 using new dangerous zone to allocate a set of  
 way-points for avoidance; 5  
 constructing a visibility graph of new navigation  
 space, in which is a set of collision free path  
 segments ;  
 linking start point to goal point by flight path  
 searching algorithm; and obtaining the terrain 10  
 profile of flight path from terrain model.

**13.** The method in claim 12, wherein

weather condition coverage area is a set of lo- 15  
 cational codes overlaying said terrain map; and  
 the weather condition can be thunderstorms,  
 windshear or any meteorological situations that  
 appear in airspace .

**14.** A method of performing real-time dynamic obsta-  
 cles avoidance comprising:

using Digitized Terrain Elevation Data (DTED) 25  
 to generate a terrain model;  
 using this terrain model to provide a navigation  
 space;  
 accessing and retrieving terrain model to gen-  
 erate a terrain map;  
 giving start and goal points on navigation space 30  
 to determine a ground track of direct flight path  
 on terrain map;  
 identifying dangerous zone based on ground  
 track and flight altitude;  
 giving an obstacles coverage area which is rep- 35  
 resented by a list of nodes;  
 adding the list of nodes of the obstacles cover-  
 age area to the list of nodes of the dangerous  
 zone;;  
 using said new dangerous zone to allocate a 40  
 set of way-points for avoidance;  
 constructing a visibility graph of navigation  
 space, in which is a set of collision free path  
 segments ;  
 linking start point to goal point by flight path 45  
 searching algorithm; and obtaining the terrain  
 profile of flight path from terrain model.

**15.** The method in claim 14, wherein

obstacles coverage area is a set of locational 50  
 codes overlaying said terrain map; and  
 obstacles can be terrain, peak, and any man-  
 made obstacles.

**16.** A method of performing real-time terrain masking  
 for terrain awareness comprising:

using Digitized Terrain Elevation Data (DTED)  
 to generate a terrain model;  
 using this terrain model to provide a navigation  
 space;  
 accessing and retrieving terrain model to gen-  
 erate a terrain map;  
 identifying dangerous zone, peaks, and obsta-  
 cles coverage area based on ground track of  
 flight path and flight altitude;  
 computing line-of-sight terrain masking; and  
 assigning color code for terrain masking and  
 awareness according to I,J,K,S parameters of  
 highlighting areas.

**17.** The method in claim 16, wherein said terrain model  
 contains a set of nodes, each represented by an in-  
 teger.

**18.** The method in claim 16, wherein

said computing line-of-sight terrain masking  
 computes the line-of-sight terrain masking with  
 respect to a selected flight altitude and heading  
 using the same locational code representa-  
 tions;  
 the dangerous zone, peaks, and obstacles rep-  
 resent a subset of nodes of terrain model and  
 are obtained from collision checking and region  
 finding with homogeneous nodes;and  
 the I,J,K parameters give the planar position  
 and elevation of a node, different color codes  
 from a mapping table are assigned to each el-  
 evation band; S representing the coverage ar-  
 ea of said node to form said masking area.

**19.** The method in claim 17, wherein

each of said nodes contain four parameters I,  
 J,K,S;  
 the four parameters I,J,K,S are obtained by  
 mapping each elevation peak point I,J,K of a  
 DTED file into a 3-D locational code of the cor-  
 responding Oct-tree;  
 parameters I, J define the 2-D coordinates of  
 planar location, with elevation K , the parame-  
 ters I,J,K define the 3-D coordinates of a space  
 location;  
 scaling factor of the elevation is introduced  
 which divides the terrain elevation into bands;  
 parameter K represents scaled elevation;  
 the nodes of said Oct-tree with equal scaled  
 value of elevation occurs in the same quadrant  
 of the corresponding quad-tree are merged;  
 scaling factor can also be non-linear, or begin  
 with a baseline in stead of the mean sea level;  
 parameter S is the coverage area of a node,  
 and is added to represent the size of a node;  
 said four parameters I,J,K,S are interleaving in

its bit positions to form an integer representation of node; and  
each terrain elevation data has its unique correspondent node.

5

20. A method of generating perspective images of terrain model, comprising:

using Digitized Terrain Elevation Data (DTED) to generate a terrain model;  
using this terrain model to provide a navigation space; and  
accessing and retrieving parameters I,J,K,S from terrain model to generate perspective images.

10

15

21. The method in claim 20, wherein said terrain model contains a set of nodes, each represented by an integer.

20

22. The method in claim 21, wherein

each of said nodes contain four parameters I, J,K,S;  
the four parameters I,J,K,S are obtained by mapping each elevation peak point I,J,K of a DTED file into a 3-D locational code of the corresponding Oct-tree;  
parameters I, J define the 2-D coordinates of planar location, with elevation K, the parameters I,J,K define the 3-D coordinates of a space location;  
scaling factor of the elevation is introduced which divides the terrain elevation into bands; parameter K represents scaled elevation;  
the nodes of said Oct-tree with equal scaled value of elevation occurs in the same quadrant of the corresponding quad-tree are merged;  
scaling factor can also be non-linear, or begin with a baseline in stead of the mean sea level;  
parameter S is the coverage area of a node, and is added to represent the size of a node;  
said four parameters I,J,K,S are interleaving in its bit positions to form an integer representation of node; and  
each terrain elevation data has its unique correspondent node.

25

30

35

40

45

23. The method in claim 20, wherein said accessing and retrieving parameters I,J,K,S comprising:  
using terrain model directly as input data without accessing the original DTED file, where the I,J,K parameters give the planar position and elevation of a node, different color codes from a mapping table are assigned to each elevation band, S representing coverage area of a node to form said perspective images of terrain.

50

55

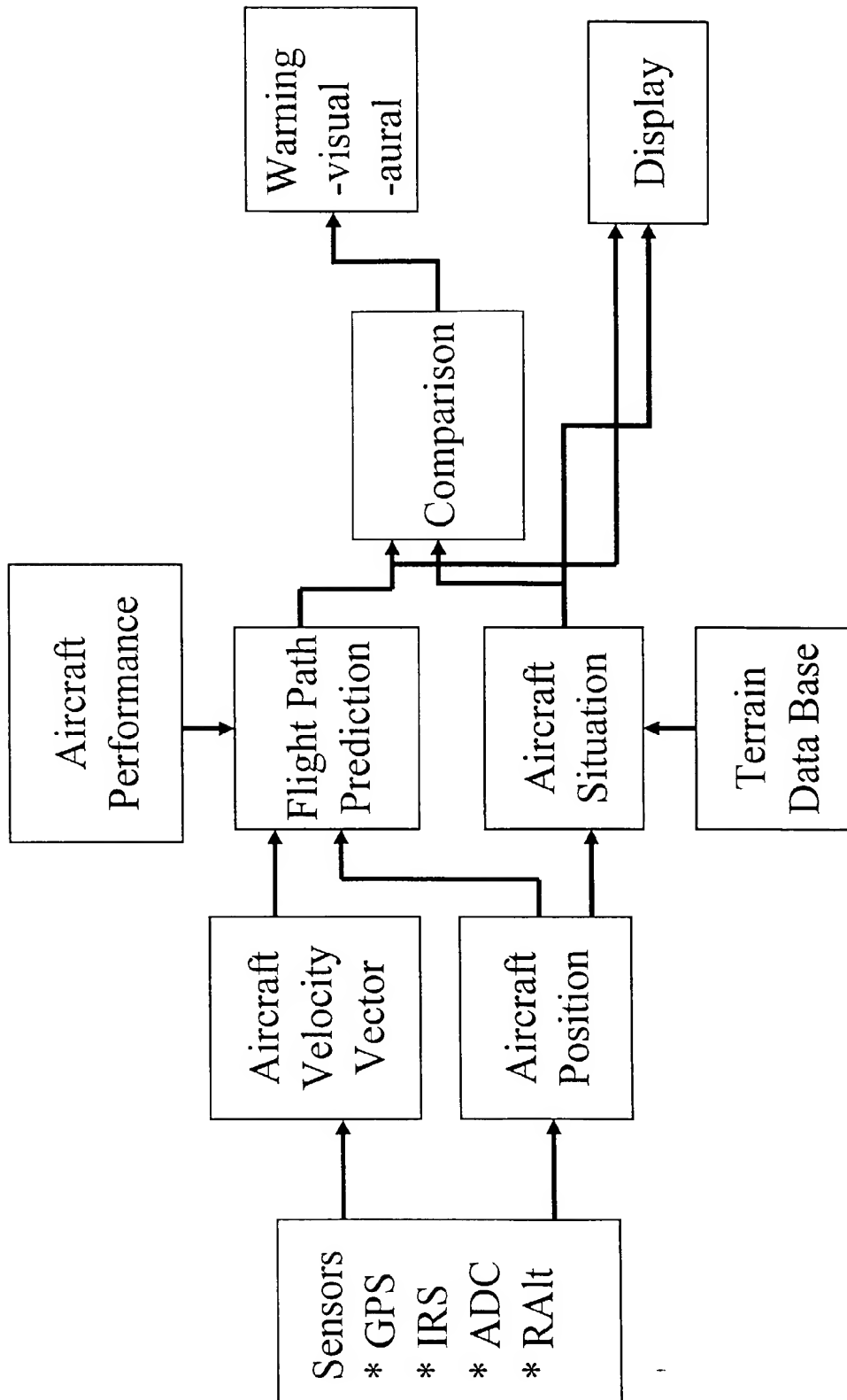
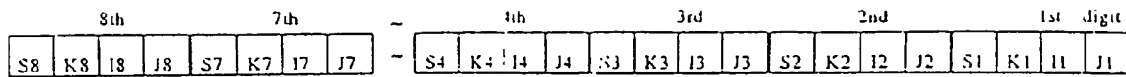


Figure 1. A schematic system diagram of a prior art of Ground Proximate Warning System



S = SIZE    K = ELEVATION    J, I = PLANAR COORDINATE

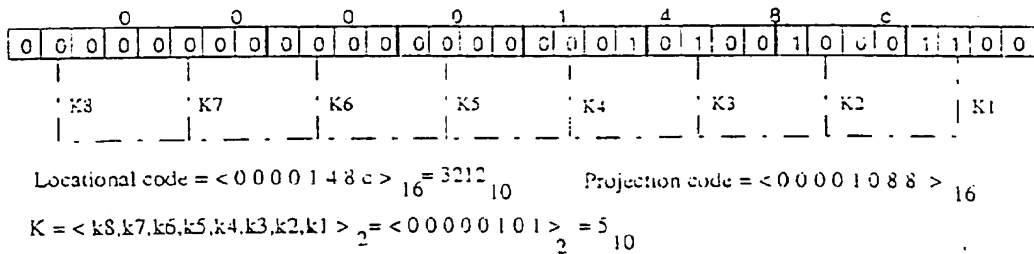
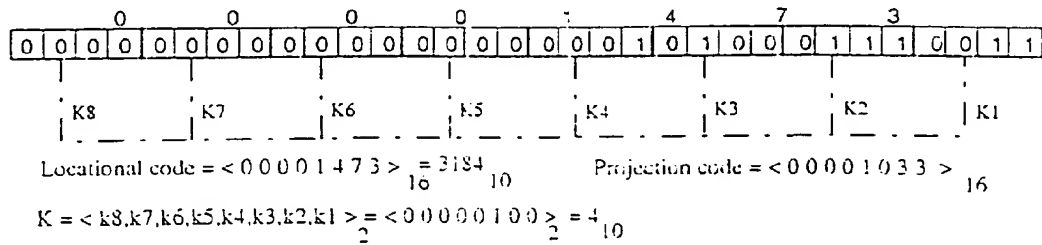


Figure 2. Bit interleave of locational code and 2-D, 3D projection.

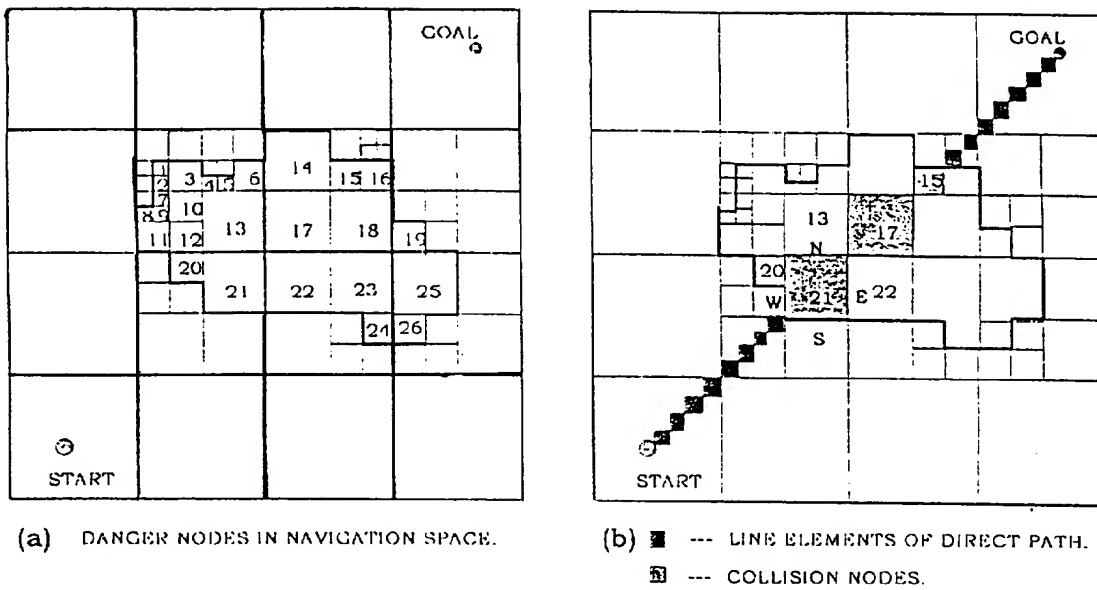


Figure 3. An example of navigation space with danger nodes.

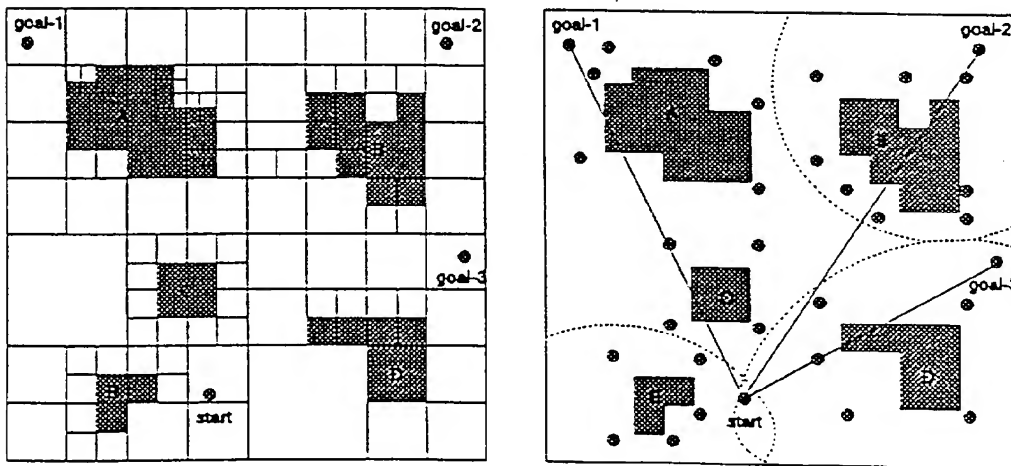


Figure 4. Possible dangerous zones with respect to heading.

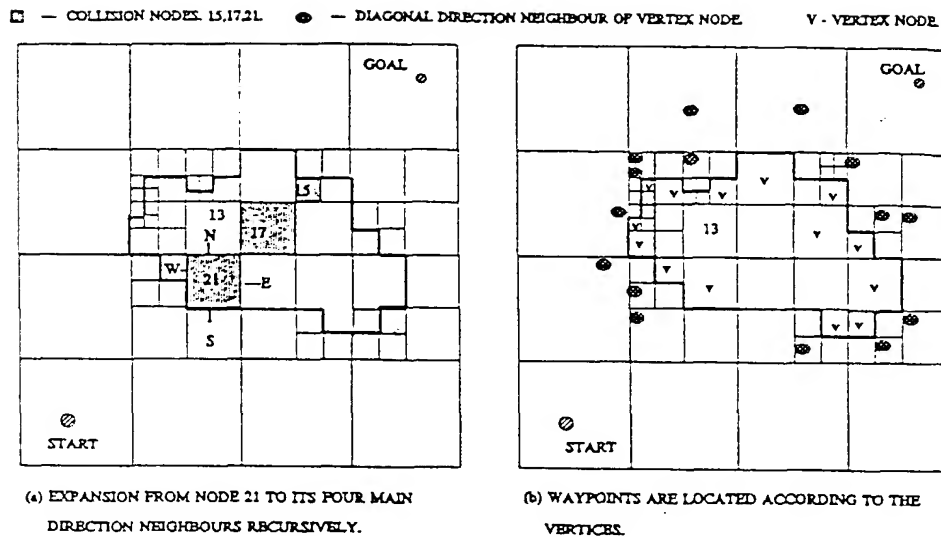


Figure 5. An example of using binary search of danger node list to determine the boundary nodes during dangerous zone expansion (dangerous zone refer to Figure 3).

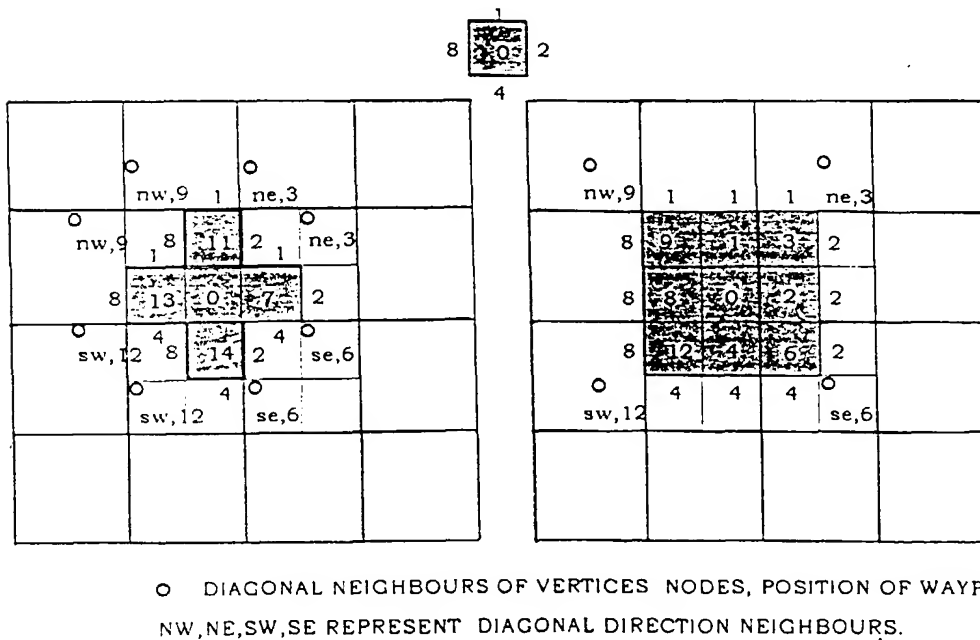
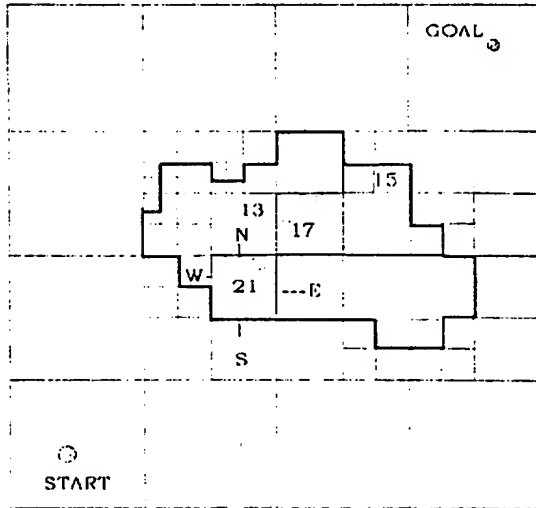
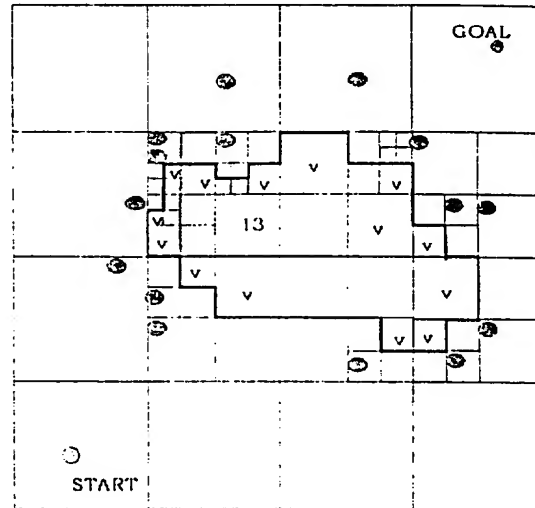


Figure 6. A combination of boundary type and waypoint position.

□ --- COLLISION NODES. 15, 17, 21 ●--- DIAGONAL DIRECTION NEIGHBOUR OF VERTEX NODE. V - VERTEX NODE.



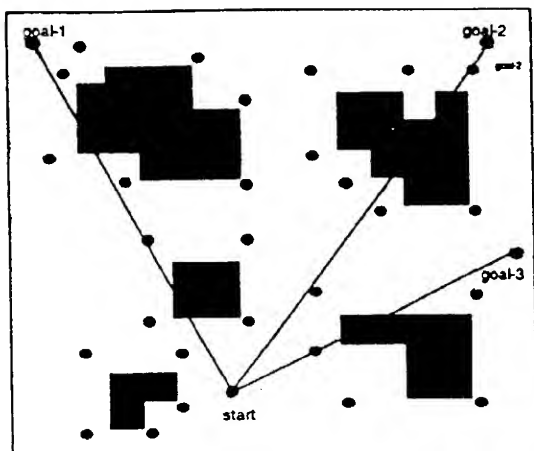
(a) EXPANSION FROM NODE 21 TO ITS FOUR MAIN DIRECTION NEIGHBOURS RECURSIVELY.



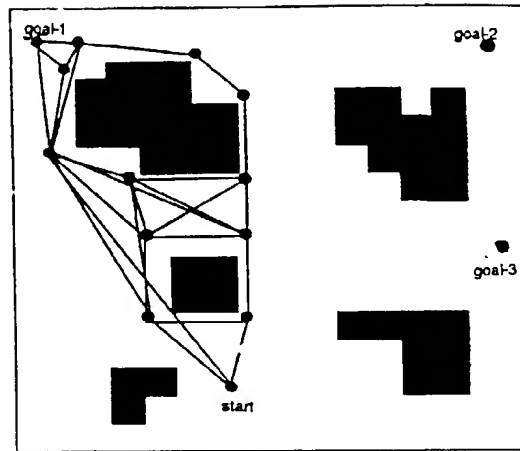
(b) WAYPOINTS ARE LOCATED ACCORDING TO THE VERTICES.

Figure 7. An example shows dangerous zone expansion and waypoints derivation.

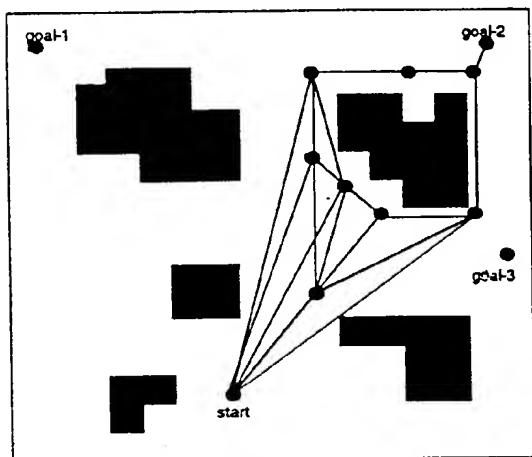




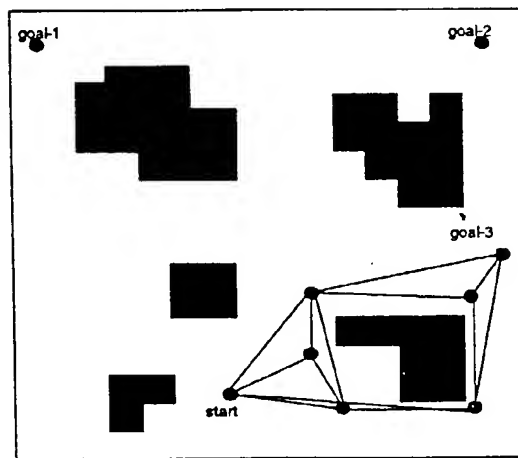
(a) A NAVIGATION SPACE WITH 32 WAYPOINTS.



(b) PATH START TO GOAL-1, 13 WAYPOINTS.



(c) PATH START TO GOAL-2, 10 WAYPOINTS.



(d) PATH START TO GOAL-3, 7 WAYPOINTS.

Figure 8. Dangerous zones with respect to different start and goal point.

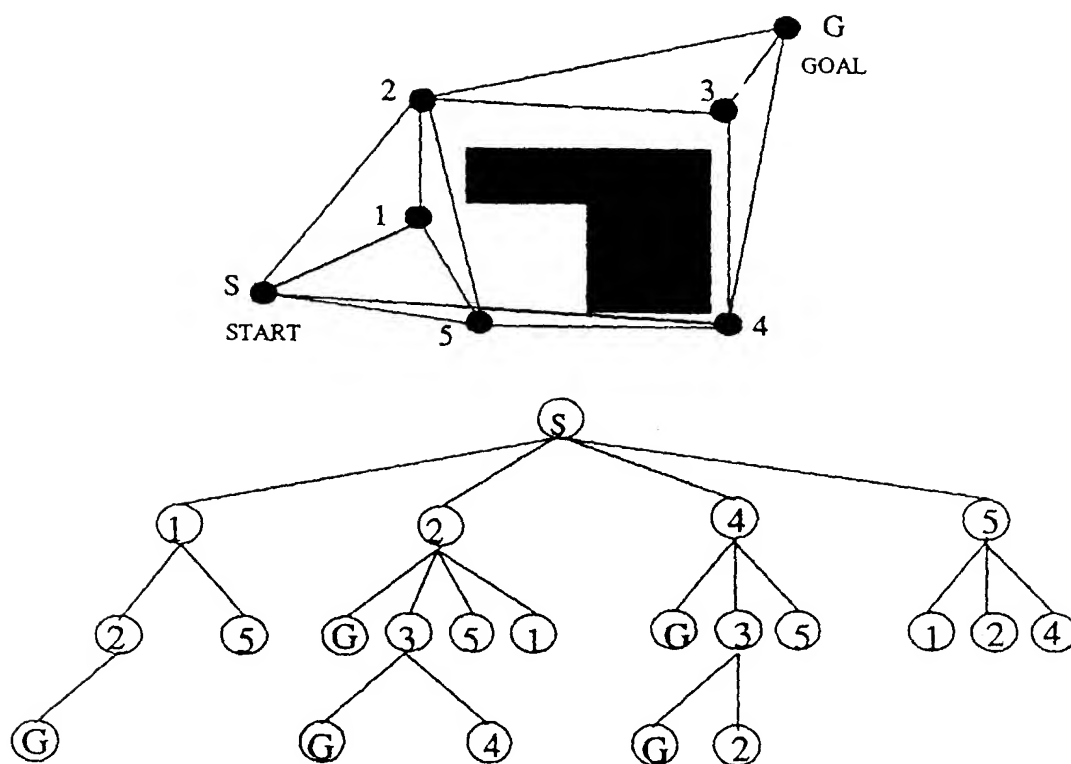


Figure 9. A visibility graph and its tree structure.

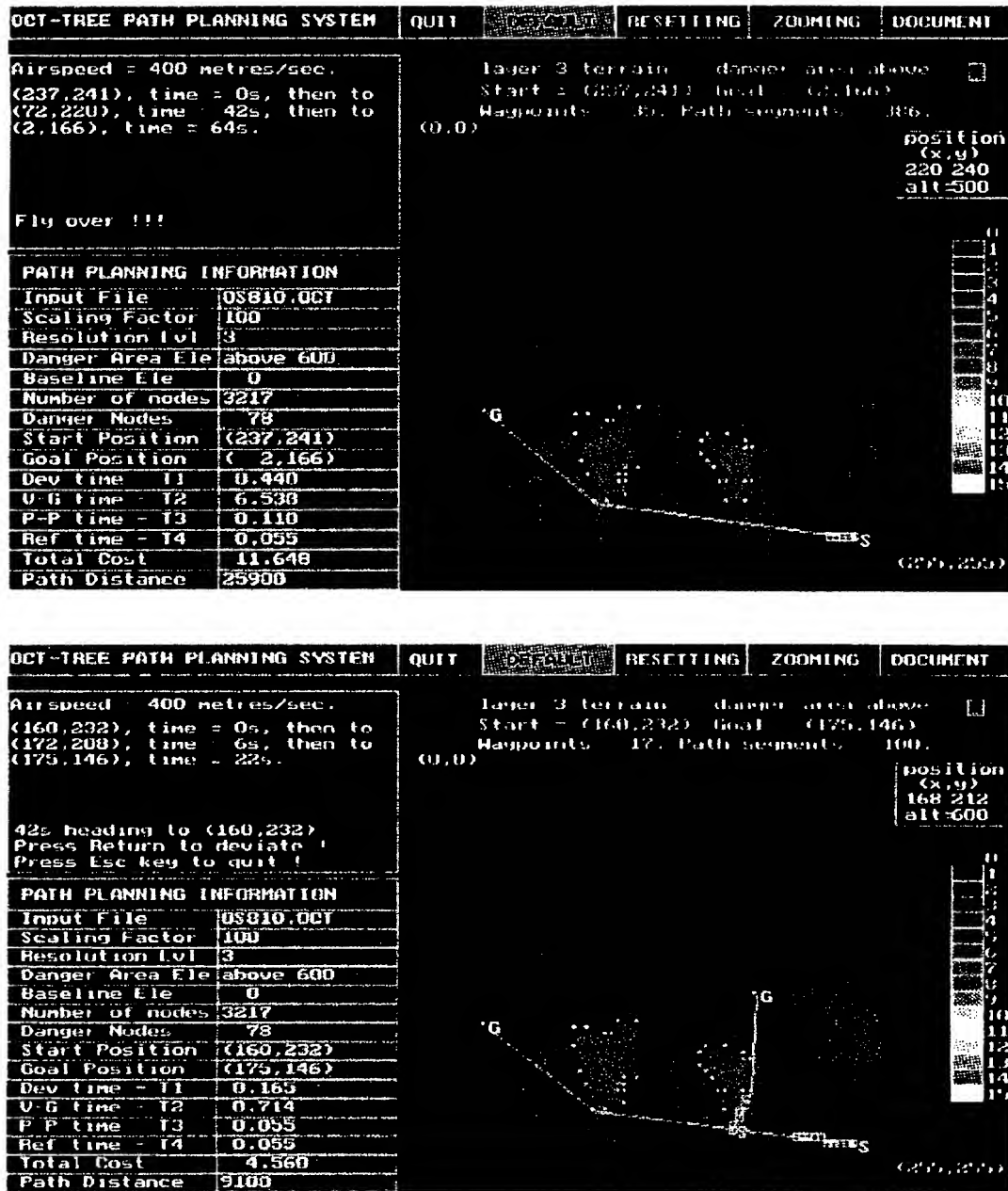


Figure 10. An example of embodiment of real time flight path planning

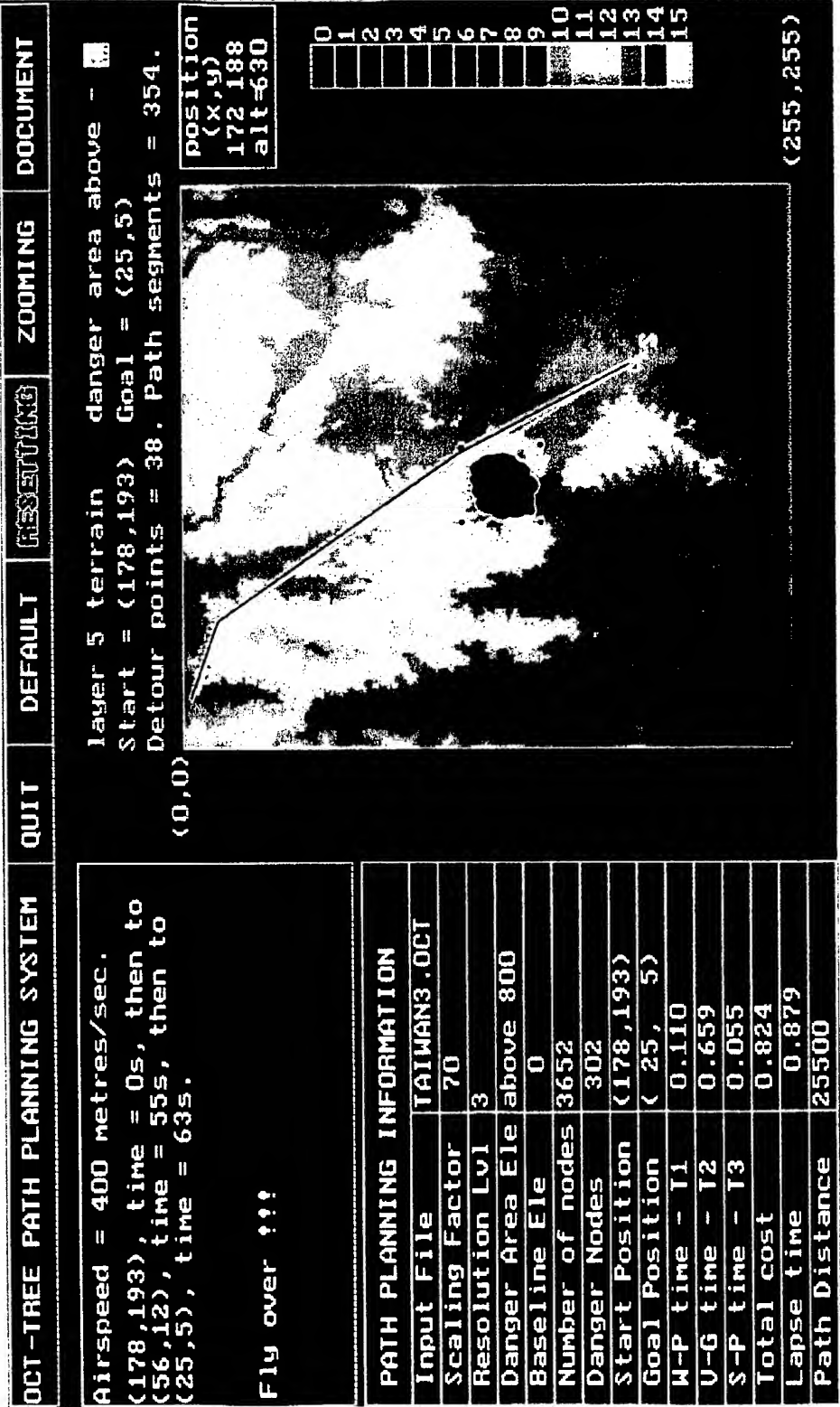


Figure 11. An example of embodiment of obstacles (weather) avoidance.

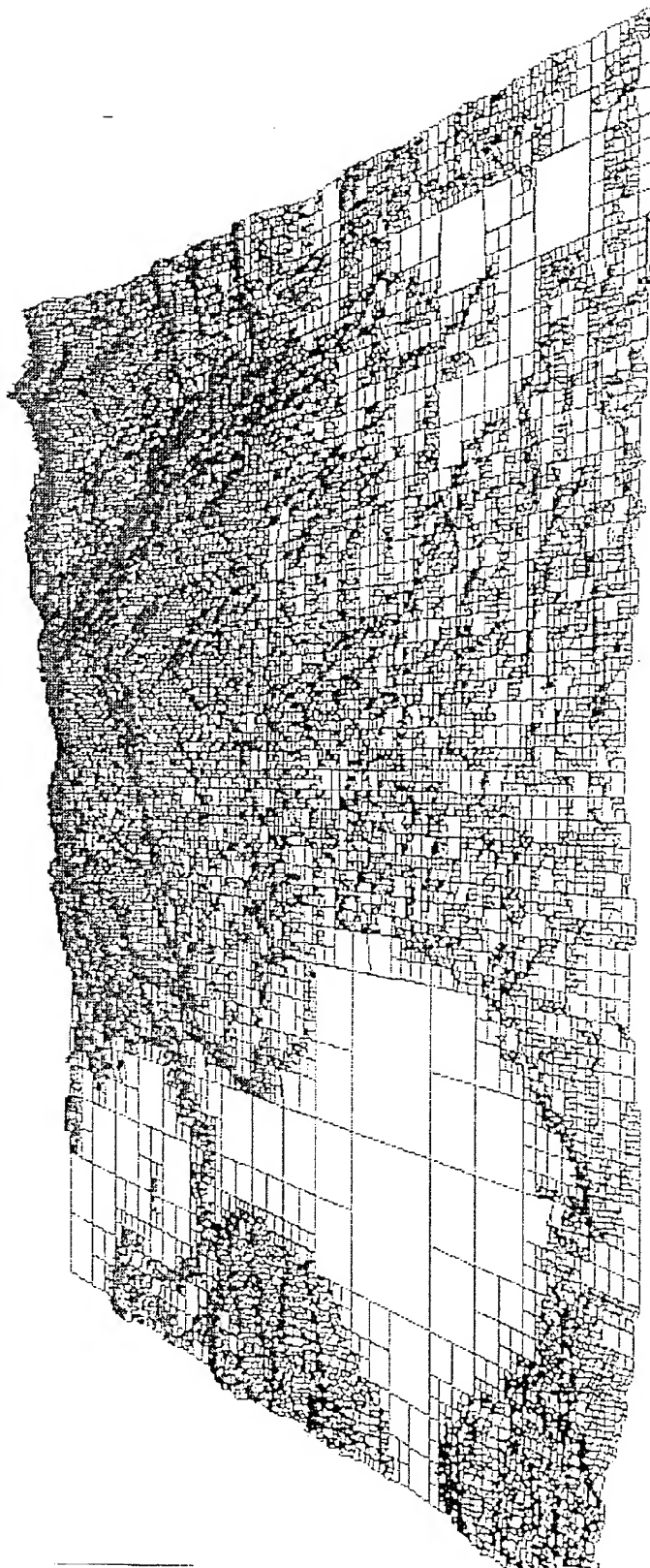


Figure 12. An example of embodiment of perspective image generating.



European Patent  
Office

## EUROPEAN SEARCH REPORT

Application Number  
EP 99 11 2442

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	ALLERTON D J: "Application of oct-trees to real-time mission management" IEE ELECTRONICS DIVISION COLLOQUIUM ON TERRAIN DATABASES AND THEIR USE IN NAVIGATION AND COLLISION AVOIDANCE; LONDON, UK APR 4 1995, no. 066, 1995, pages 3/1-3/8, XP002124293 IEE Colloq Dig; IEE Colloquium (Digest) 1995 IEE, Stevenage, Engl * the whole document *	1-23	G01C21/00 G08G5/04
X	ALLERTON D J ET AL: "Oct-tree terrain modelling methods for terrain reference navigation systems" AERONAUT J; AERONAUTICAL JOURNAL MAY 1996, vol. 100, no. 995, May 1996 (1996-05), pages 157-164, XP000856843 * the whole document *	1-23	
A	US 5 257 347 A (BUSBRIDGE MICHAEL L ET AL) 26 October 1993 (1993-10-26) * abstract *	1	TECHNICAL FIELDS SEARCHED (Int.Cl.7)
A	EP 0 655 698 A (HONEYWELL INC) 31 May 1995 (1995-05-31) * abstract *	1	G01C G08G
The present search report has been drawn up for all claims			
Place of search <b>THE HAGUE</b>		Date of completion of the search <b>30 November 1999</b>	Examiner <b>Hoekstra, F</b>
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

EPO FORM 1503 03.82 (P04C01)

**ANNEX TO THE EUROPEAN SEARCH REPORT  
ON EUROPEAN PATENT APPLICATION NO.**

EP 99 11 2442

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on  
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

30-11-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5257347 A	26-10-1993	GB 2187616 A, B	09-09-1987
EP 0655698 A	31-05-1995	US 5504686 A	02-04-1996
		EP 0655714 A	31-05-1995
		EP 0655699 A	31-05-1995
		IL 111813 A	08-02-1998